**Laporan Proyek Akhir NLP – Tweet Sentiment Analysis using BiLSTM**

**Link Presentasi:** <https://www.canva.com/design/DAGbILf8Gp0/4XuV5G_sYFCczELdR_XzRw/edit?utm_content=DAGbILf8Gp0&utm_campaign=designshare&utm_medium=link2&utm_source=sharebutton>

**Anggota kelompok:**

1. 2602059645 - Kevin Purnomo
2. 2602168406 - Raphaelle Albetho Wijaya
3. 2602204090 - Sheraldo Halim

**Introduction**

Sentiment analysis adalah suatu bidang di dalam Natural Language Processing (NLP) yang bertujuan untuk menganalisis sentimen dari suatu teks. Sentimen tersebut bisa bersifat positif, negatif, netral, dan lain-lainnya. Di dalam penerapan nyata, Sentiment analysis dapat digunakan untuk menganalisis sentimen dari suatu topik di dalam media sosial, mengetahui kepuasan pelanggan di aplikasi e-commerce, dan lain-lainnya. Di dalam proyek yang kami kerjakan, kami menerapkan Sentiment analysis untuk mengenalisis sentiment dari suatu tweet (diambil dari aplikasi Twitter/X). Dataset yang kami gunakan diambil dari Kaggle, yang dimana data tersebut akan digunakan untuk melatih model BiLSTM. Model tersebut kemudian akan digunakan untuk deployment, yang dimana user bisa memasukan sebuah tweet, kemudian aplikasi tersebut akan mengklasifikasikan sentimen (positif, negatif, dan netral) dari tweet tersebut.

**Data Acquisition**

Pada tahap ini, kami mengambil dataset tweet beserta sentimen dari Kaggle (<https://www.kaggle.com/datasets/jp797498e/twitter-entity-sentiment-analysis>). Dapat dilihat bahwa datasetnya belum sepenuhnya bersih, masih terdapat data duplikat dan beberapa missing values yang harus ditangani. Proses tersebut akan ditangani pada tahap data preprocessing

**A screenshot of a computer

Description automatically generated**

**Text and Data Preprocessing**

Berikut adalah pendekatan kami untuk menangani kedua masalah tersebut: 

Setelah code diatas dijalankan, semua data duplikat dan missing values berhasil ditangani: A screen shot of a computer

Description automatically generated

Berikut adalah data unik yang terdapat di dalam label ‘sentiment’ beserta jumlah row-nya: A black background with white text

Description automatically generated

Dari segi persebarannya, label ‘sentiment’ tersebut tergolong balanced.

Sebelum melatih model, kita harus melakukan tahap preprocessing pada data tweet yang terdapat di dalam dataframe tersebut. Berikut adalah codingan yang saya gunakan:

A computer screen with text

Description automatically generated

Di dalam fungsi tersebut, beberapa tahap yang saya lakukan kepada teks tweet adalah sebagai berikut:

1. Mengubah teksnya menjadi lowercase, agar kata yang uppercase tidak tergolong sebagai kata yang berbeda dengan kata yang lowercase
2. Menghapus URL, tanda baca, dan angka karena tidak bermakna sama sekali terhadap sentimen
3. Menghapus emoji untuk mengurangi noise dan untuk mengurangi ukuran vocabulary
4. Menghapus spasi berlebih untuk menghindari masalah saat tokenization
5. Melakukan normalisasi untuk mengurangi feature space, membantu model mengenali beberapa variasi kata dari suatu kata yang bermakna sama

Setelah itu, data tweet yang sudah di preprocess akan melewati proses Lemmatization dan penghapusan stop words.

A screenshot of a computer screen

Description automatically generated

Lemmatization cukup penting karena dapat mengubah beberapa variasi kata, seperti kata dengan imbuhan menjadi bentuk kata dasar. Hal tersebut dapat berguna untuk mengurangi dimensi data dan menyeragamkan kata ke bentuk dasarnya. Mengapa kami menggunakan lemmatization dan bukan stemming? Hal tersebut kami lakukan karena dataset yang kami gunakan tidak besar, sehingga masih memungkinkan untuk menggunakan lemmatization, tidak hanya itu saja, lemmatization menggunakan kamus linguistik untuk mempertimbangkan arti kata dan konteksnya, yang dimana dapat menghasilkan kata dasar yang lebih akurat dibandingkan dengan stemming . Hal kedua yang akan dibahas adalah penghapusan stopwords (the, is, at, dll). Stopwords harus dihapus karena mengandung kata-kata umum yang tidak memiliki makna yang penting, dengan dihapusnya stopwords, pemrosesan dapat berjalan dengan lebih efisien karena dapat fokus ke kata-kata yang lebih bermakna. Disini, saya tidak menghapus no/not (kedua kata tersebut terdapat di dalam stopwords) karena kedua kata tersebut dapat memberikan informasi yang penting berkaitan dengan sentiment tweet.

Tahap berikutnya adalah melakukan encoding pada label di dalam dataset tersebut. Berikut adalah pembagiannya:

* 'Negative': 0,
* 'Positive': 1,
* 'Neutral': 2

Disini, saya drop semua baris yang memiliki sentiment ‘Irrelevant’ (Tweet tidak berkaitan dengan entity) sebab sentiment tersebut bisa berkonotasi positif, negatif, dan netral, yang dimana kami tidak memerlukan kolom ‘entity’ di dalam Sentiment analysis. Oleh karena itu, ada baiknya data dengan sentiment ‘Irrelevant’ dihapus.

Setelah itu, tahap terpenting di dalam NLP akan dilakukan, yaitu Text Tokenization, berikut adalah code yang saya gunakan untuk tokenisasi data tweet.

A computer screen shot of text

Description automatically generated

Di dalam code tersebut, terdapat dua proses penting yang saya lakukan disamping tokenisasi. Yang pertama adalah text to sequences, proses tersebut berfgungsi untuk mengubah teks menjadi urutan angka berdasarkan tokenisasi yang sudah dilakukan. Yang kedua adalah pad sequences, hal tersebut dilakukan untuk memastikan semua sequences yang didapat dari proses pertama memiliki panjang yang sama. Hal tersebut cukup penting untuk pemodelan BiLSTM karena membutuhkan dimensi input yang konsisten. Setelah dilakukan kedua proses tersebut, ada baiknya untuk menyimpan tokenizer tersebut ke dalam suatu file yang dapat digunakan kembali pada proses deployment.

A black background with white text

Description automatically generated

**Model Training**

Sebelum melatih model, kami akan melakukan splitting data terlebih dahulu.

A screen shot of a computer

Description automatically generated

Dari data yang sudah displit, proses pemodelan dapat dikerjakan

A screen shot of a computer code

Description automatically generated

Berikut adalah penjelasan dari masing-masing layer di dalam model BiLSTM yang kami buat:

1. Embedding layer: Berfungsi untuk mengubah padded sequences menjadi representasi vektor. Hal tersebut dapat membantu model memahami hubungan semantik antar kata dalam ruang vektor
2. BiLSTM layer: Berfungsi untuk memproses data (sequence) secara dua arah (kedepan dan kebelakang) untuk memahami konteks data secara lebih baik sebab bekerja secara dua arah. Pada layer BiLSTM pertama, return sequence dibutuhkan karena diikuti oleh layer BiLSTM kedua yang membutuhkan input dalam bentuk urutan langkah waktu.
3. Dropout layer: Berfungsi untuk menonaktifkan neuron sejumlah 30% atau 0.3 yang terdapat pada layer sebelumnya. Hal tersebut dilakukan untuk mencegah overfitting
4. Dense Layer: Berfungsi untuk menangani non-linearitas dan membantu model dalam mempelajari pola yang kompleks, di dalam struktur model tersebut. Dense layer dapat mengubah representasi dari BiLSTM menjadi fitur yang terstruktur. Aktivasi ReLU digunakan untuk mengurangi kemungkinan terjadinya *Vanishing Gradient Problem* dan meningkatkan efisiensi dari komputasi*.* Pada output dense layer, terdapat 3 neuron yang mewakili 3 sentimen di dalam dataframe (positif, negatif, dan netral) dan menggunakan aktivasi Softmax. Kami menggunakan Softmax karena paling efektif dalam multiclass classification task (lebih dari 2 class), terutama dalam konteks Sentiment analysis.

Kami compile model tersebut menggunakan Adam optimizer dengan learning rate sebesar 0.005 (Tidak terlalu cepat dan tidak terlalu lambat), fungsi loss sparse categorical crossentropy (cocok untuk multiclass classification dan tipe data integer), dan metrik akurasi.

Berikut adalah struktur dari model BiLSTM yang dibuat:

A screenshot of a computer program

Description automatically generated

Setelah itu, kami lakukan proses training dengan 30 epochs.  
A computer screen shot of a code

Description automatically generated

Setelah proses pembuatan model dan training model, model tersebut akan disimpan ke dalam file dengan format h5 agar bisa digunakan pada saat deployment.



**Model Evaluation**

Setelah menjalani proses training, proses berikutnya adalah evaluasi model, berikut adalah hasilnya:

* Training accuracy: 0.9711
* Test Accuracy: 0.9107
* Validation accuracy: 0.9336

A graph of a line graph

Description automatically generated with medium confidenceA graph with orange lines and blue lines

Description automatically generated

A diagram of a test

Description automatically generated with medium confidenceA red and blue squares with white text

Description automatically generated

**Model Deployment**

Dari segi deployment, kami menggunakan Streamlit sebagai sarana deployment. Berikut adalah tampilannya:

A screenshot of a computer

Description automatically generated

**Kesimpulan**

Dari laporan yang kami buat, dapat dilihat bahwa terdapat banyak sekali penerapan NLP di kehidupan nyata, salah satunya adalah Sentiment analysis yang juga terdapat banyak use case. Di dalam proses pengembangan model NLP, tahapan yang harus dilakukan adalah sebagai berikut: Data Acquisition, Text and Data Preprocessing, Model training, Model Evaluation, dan Model Deployment. Pada tahap pemodelan, kami memilih BiLSTM karena model tersebut memiliki rasio performa dan biaya komputasi yang cukup baik dibandingkan dengan transformers. Menurut kami, model BiLSTM sudah cukup baik untuk task Sentiment analysis. Pada tahap evaluasi model, dapat dilihat bahwa tidak terjadi overfitting, sebab masing-masing data training, testing, dan validation memiliki akurasi yang baik. Dari segi deployment, pengguna dapat menginput suatu tweet dan aplikasi tersebut akan memberikan output yang berisi sentimen dari tweet tersebut dan confidence model akan prediski tersebut.